

Animation of multi-flexible body systems and its use in control system design

Carl Juengst
Boeing Defense & Space Group
P.O. Box 3999, Seattle, WA 98124

Ron Stahlberg
Center for Simulation and Design Optimization
of Mechanical Systems
University of Iowa
Iowa City, IA 52242

Introduction

Animation can greatly assist the structural dynamicist and control system analyst with better understanding of how multi-flexible body systems behave. For multi-flexible body systems, the structural characteristics (mode frequencies, mode shapes, and damping) change, sometimes dramatically with large angles of rotation between bodies. With computer animation, the analyst can visualize these changes and how the system responds to active control forces and torques. Figure 1 characterizes the type of system we wish to animate.

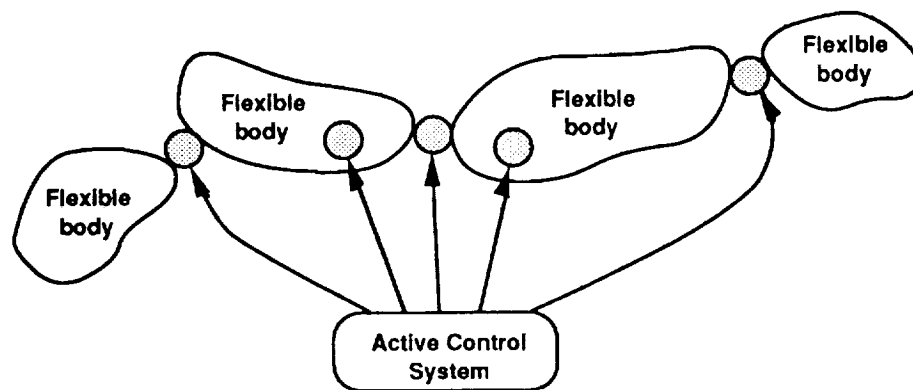


Figure 1. Multi-Flexible Body System To Be Animated

The lack of clear understanding of the above effects was a key element leading to the development of a multi-flexible body animation software package. The resulting animation software is described in some detail here, followed by its application to the control system analyst. Other applications of this software can be determined on an individual need basis.

A number of software products are currently available that make the high-speed rendering of rigid body mechanical system simulations possible. However, such options are not available for use in rendering flexible body mechanical system simulations. The desire for a high-speed flexible body visualization tool led to the development of the Flexible Or Rigid Mechanical System¹ (FORMS) software. This software was developed at the Center for Simulation and Design Optimization of Mechanical Systems at the University of Iowa. FORMS provides interactive high-speed rendering of flexible and/or rigid body mechanical system simulations, and combines geometry and motion information to produce animated output. FORMS is designed to be both portable and flexible, and supports a number of different user interfaces and graphical display devices. Additional features

have been added to FORMS that allow special visualization results related to the nature of the flexible body geometric representations.

History

The FORMS software is a direct descendent of the Visualization of Dynamic Systems² (VDS) software also developed at the University of Iowa. This software was designed to perform high-speed animation of rigid body simulations. VDS afforded the user a high degree of interactive control of simulation parameters, such as color, visibility, and viewing orientation. This software utilized euler vectors to specify body position and orientation, which combined rotational and translational information into one data component. The rigid nature of the bodies being rendered allowed VDS to support additional features, such as interference checking and casting of shadows. The input formats supported by VDS were somewhat limited, using an internally developed proprietary format. Translators were, however, provided between this format and a number of standards such as the Initial Graphics Exchange Specification³ (IGES) and Movie.BYU⁴. The structure of VDS allowed for an open architecture, which enabled the software to support a large number of different user interfaces and display devices. Any combination of user interfaces and display devices could then be supported on a particular platform.

The development of FORMS incorporated many of the features utilized in the development of VDS, and also allowed for the addition of a significant number of new capabilities. FORMS is compatible with the VDS software, with the exception of some features directly related to the rigid nature of VDS objects, and can be used to produce animations from VDS input files without modification. The limitation of the proprietary input format imposed by VDS was also lifted, as FORMS supports three different input formats: the VDS proprietary format; a close modification of the VDS proprietary format intended to support flexible body information; and an ASCII input format designed to be easily edited and modified. The flexible nature of the bodies being rendered also required that the techniques for storing both geometry information and motion data be updated. Additionally, FORMS was intended to support a number of different modes of interaction, including structure editing and harmonic motion display.

Organization

The FORMS software uses a hierarchy to store information required to produce an animation sequence. Three different types of information are stored in this hierarchy. The first is what is considered essentially universal data. Data of this type influences the animation as a whole and includes such standard rendering options as view specifications, lighting, and environmental parameters. The second type of information stored in the hierarchy involves the "objects" or "parts" of the mechanical system being rendered. The nature of the motions specified in FORMS allows a linked list to be used in storing such objects. The use of a linked list rather than a tree-like structure is made possible by the fact that motions for the flexible bodies are represented as displacements on individual nodes and not motions that can be applied to a group of objects. Each object in this list contains an associated geometric representation for the object and object-specific attributes such as color and representation (i.e., solid, wireframe, point). The final type of information stored in the hierarchy is animation data. This data is stored on a per-frame basis and can be of one of two types: euler vectors, which are converted into displacement information for each node in the geometry, and straight nodal displacement data. In the context of each individual frame, the animation data is then organized according to the order specified by the object list in the hierarchy.

The functional organization of the FORMS software was designed to maximize flexibility and portability. There are three main components to the FORMS software: a communication manager, display device drivers, and user interfaces. The main FORMS executable functions as the communications manager. It handles requests from the various user interfaces and then instructs the display devices to update their representations accordingly. This organizational model is illustrated in figure 2.

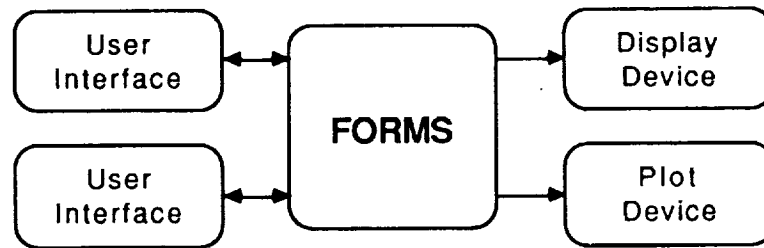


Figure 2. FORMS Communication Channels

The actual FORMS executable is isolated from any individual user interface or device driver, a design that greatly improves the portability of the FORMS software.

Data Requirements

Three types of data are required as input to the FORMS software: geometric data, motion data, and initial configuration data. Geometric data can be in the form of either rigid or flexible body representations. For rigid bodies, the data is in the form of a polygonal mesh. Flexible bodies require that geometric information be presented in the form of a finite element mesh. At the current release, seven different types of finite elements are supported by the FORMS software: point, beam, linear triangle, linear quadrilateral, linear tetrahedron, linear solid wedge, and linear solid. The representations of these elements are as presented in the IGES specification.

Motion information is specified as either euler vectors or nodal displacements. The type of motion information allowed depends on the associated geometric representation for the object. If the object is a rigid body, then only euler vector data can be used in positioning the object. For flexible objects, motion information can be specified as either nodal displacements or as a combination of euler vectors and nodal displacements. If a combination of euler vectors and nodal displacements is supplied, then the euler vectors are first converted to nodal displacements. After this conversion, the individual nodal displacements specified as input are applied. This "two-step" motion is critical in analyzing certain classes of problems.

Initial configuration information is supplied in the form of a definition or ".def" file. This file contains initial viewing, lighting, and environmental parameters, as well as initial object attributes and motion data ordering. Each object represented in the animation has a corresponding entry in the definition file. This entry contains the object's initial color, representation, shading model, and geometric representation. Geometric representations are denoted by name, which must correspond to the name of a representation supplied as input in one of the geometry files. Motion data ordering is specified in the form of an ordered list of objects. Additional objects may appear in the definition file for which no motion is specified. These objects are considered stationary.

The formats for the required data are of one of three types: VDS module format, FORMS flexible module format, or ASCII "easy" file format. The module format and flexible module formats are

proprietary file formats, which can be in either a space-reducing, machine-dependent binary format or a platform-independent ASCII representation. Both of these formats can be used to encode both motion and geometric data, but are not readily editable because they contain additional information to speed file access. In order to facilitate conversion to these formats, a number of translators from various standards are provided, including IGES, Movie.BYU, and DADS-3D⁵. The "easy" file format is an ASCII format designed to be edited using any standard text editor. The format of these files is relatively simple. No strict layout is required with regard to column location, line spacing, etc. In this way, the output from nearly any package can be manipulated into the "easy" file format with a minimum of effort.

Rendering

The major concern in rendering flexible objects is that as nodal displacements occur, faces of finite elements that were initially essentially planar lose this property. Therefore, in order to render flexible bodies with sufficient accuracy and still maintain the required throughput for animation, some pre-processing is required. In FORMS, this pre-processing takes the form of triangulation of all polygons and/or finite element faces with more than three vertices. This division of such surfaces into triangles allows each face to be decomposed into a number of planar facets. Since each three nodes will uniquely determine a plane, each triangular facet can then be rendered as planar and still give the appearance of the required flexible motion. This process is illustrated in figure 3.

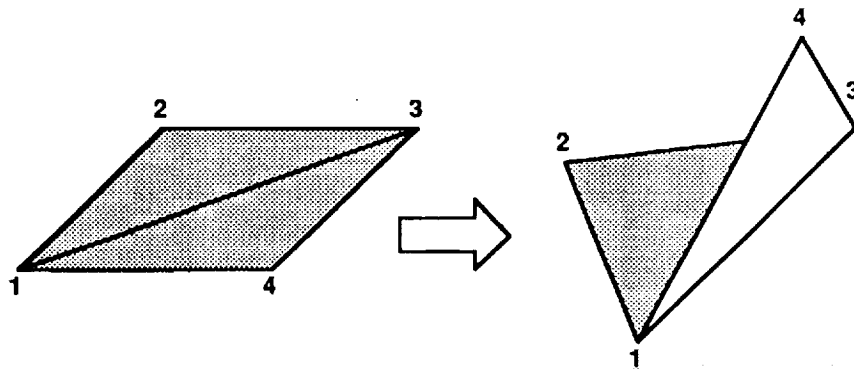


Figure 3. Triangulation into Planar Facets for Flex

Since the facets are assumed to remain contiguous, the representation will remain accurate. For simple finite element representations, the triangulation is straightforward. For more advanced polygonal representations, a modified version of the triangulation algorithm presented by Feng and Pavlidis⁶ is used; the only restriction is that the polygons be simple. The resulting triangulation is used further in graphically editing the objects, since a strictly triangular representation of the scene allows for a general purpose routine to test for intersection between objects and a selected point. By computing the triangulations of these objects prior to animation and independent of the displacements involved, rendering requires only the updated positions of the individual nodes to be computed prior to drawing. The mesh generated in triangulating an object can be displayed optionally on a per-object basis.

The viewing and lighting attributes specified for each animation are generic in nature and are not of particular interest to this paper. The only aspect of viewing that will be mentioned is the fact that FORMS supports multiple views of an individual animation. Currently, up to four synchronized, independent views of any animation are permitted.

Environmental parameters are of more interest, because some values are directly related to the flexible nature of the bodies being rendered. Values that relate to the animation in general include background color and interval control. Interval control allows for strict frame-rate control, as opposed to allowing the platform to render as fast as possible. The parameters related to the flexible nature of the objects include: the color used in rendering the triangle mesh generated during preprocessing; the colors used in representing active vs. inactive and active vs. special elements during editing; and the period used in displaying harmonic motion. Each of these parameters will be discussed in more detail as they relate to individual options.

By far the greatest number of attributes are possessed by the individual objects. These parameters relate either to the general appearance of the object or to specific flexible body options. The parameters that affect general appearance include color (including diffuse, ambient, and specular color, if supported), shading model (flat or Gouraud), label status, position, and physical representation. Current representation options include solid, wireframe, point (only vertices or nodes are rendered), stick (a combination of wireframe and point representations), and disabled. Additional parameters allow for specific flexible body options. These parameters include:

superimposition	Controls display of the undeformed geometry superimposed over the current representation. Additionally, the color and representation attributes for the superimposed geometry can be independently specified.
labels	Allows the rendering of node and/or element labels.
pointsize	Controls the size of nodes rendered in point or stick representation.
beamwidth	Controls the size of the lines rendered in wireframe or stick representation.
active vs. inactive elements	Allows any element or range of elements to be made inactive, and causes them to not be displayed.
trace nodes	Causes any element or range of elements to leave a "smoke trail" as it moves from frame to frame.
inferior representations	Allows elements that do not have the representation specified for the object to be viewed. In some representations, such as solid, certain element types have no corresponding representation (i.e., beam or point elements). This option causes such elements to be rendered in their native representations.
mesh representation	Enables display of the triangle mesh generated during preprocessing of this object.

Any or all of these parameters can be manipulated interactively for any or all objects.

Animation Control

A number of controls are available with respect to the display of an animation sequence. These include the display rate, direction, sequence, and increment. Frames are normally displayed at a rate determined by the limitations of the current hardware platform, but they can also be set to appear at a pre-determined rate, as long as it does not exceed the capabilities of the hardware. The order in which frames are displayed is totally under user control, with any sequence, direction, or group of individual frames being valid for display. In the event that different timestep increments are desired for display other than those available from the input data, an increment value is available for use in displaying frame sequences. If required, linear interpolation is performed to generate intermediate frames.

It is also desirable at times to magnify the motions or displacements for a certain object. In this case, an individual motion magnification factor is maintained for each object. Separate magnification factors are kept for nodal displacement values and euler vector values. Nodal displacement values may be magnified by individual factors in any of three dimensions (x, y, z). Euler vectors can be magnified for any of seven values (tx, ty, tx, er, ex, ey, ez).

A final animation option is provided in harmonic motion display. In this mode, a deformed state of the simulation is chosen as the base frame. FORMS then uses this frame as an ending position and the undeformed geometry as an initial position. It then interpolates between the two frames, based on the number of frames specified for the period of harmonic motion. The number of frames is specified as an environment parameter. In this mode, the total number of frames available for display is updated to reflect the limitations of the harmonic motion.

Editing

In order to make the information displayed during an animation more useful, it is often desirable to change the appearance of certain elements or disable them altogether. In FORMS, these types of operations are accomplished through the use of a special rendering mode. In this mode, individual elements of a flexible body can be selected with the aid of a mouse or other pointing device. Depending upon the values set when initiating edit mode, selected elements will either be toggled between active and inactive status or active and special status. In active vs. inactive status, inactive elements are no longer displayed when returning to normal display mode. However, in edit mode, inactive elements are displayed in a distinct color to signify their modified status. In active vs. special status, individual elements may be assigned distinct color, representation, pointsize, and beamwidth values. These new values will then be used in rendering the special elements after returning to normal display mode. As with active vs. inactive status, special elements are displayed in a distinct color from other active elements while editing.

Other Information

Aside from the qualitative information provided through the animation of mechanical system motion, more quantitative information is often desired from any analysis tool. To provide such data, FORMS has two additional output options: X-Y plotting and parameter display. The X-Y plotting option is performed in a separate window from the animation. Plotting occurs in synchronization with the animation and simultaneously provides both a quantitative and qualitative view of the system under study. The range of expressions available for plotting includes any system variables, such as nodal displacements, euler vectors, and frame count. Other available

options include incremental counters, standard library functions, and any numeric constants. The range and scale of both axes can be user-specified, as can the color of the resulting plot. Plot results can be superimposed to compare various data analyses.

The "easy" file format also provides for user-specified parameter data, other than the standard nodal displacements and euler vectors. The parameter data can be entered in the motion data file and then displayed during the playback of an animation in a separate window. In this way, FORMS can inform the user of significant events or non-numeric results.

User Interfaces/Device Drivers

As was mentioned earlier, FORMS was designed with portability and flexibility in mind. To that end, two different user interfaces are currently available. The first is a standard command line keyboard interface. The second is a graphical user interface built on top of OSF Motif⁷. The graphical user interface uses a number of specialized widgets to facilitate FORMS operations.

FORMS also currently supports two different graphical display devices. Drivers currently exist to support either X11⁸ or Silicon Graphics GL⁹.

Multi-Flexible Body Simulations

The FORMS software requires animation data containing the rigid rotation and translation of each body and/or the displacements of the nodes for flexible bodies, as described above. This data can be generated by running multi-flexible body dynamics simulations such as TREETOPS¹⁰, DISCOS¹¹, SADACS^{12, 13}, or any other simulation that can compute rigid body motion and/or flexible node displacements. We are currently using the Spacecraft Appendage Dynamics And Control Simulation (SADACS) as the driver for FORMS. SADACS was derived by Boeing Defense & Space Group for faster simulation of large-angle, flexible body vehicles. SADACS utilizes SD/EXACT¹⁴ rigid body dynamics code run in parallel with the FB2 code (linearized flexible dynamics updated at user-defined increments). The dynamics are driven by a control system and detailed actuator and sensor hardware models, including friction and other nonlinear effects. A simplified SADACS block diagram is shown in figure 4.

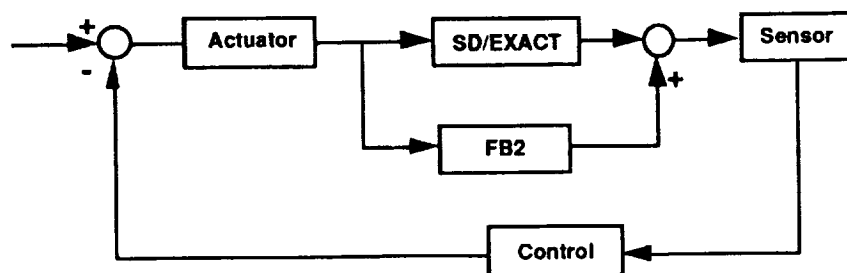


Figure 4. Simplified SADACS Block Diagram

The animation data is written to an ASCII file in the FORMS "easy" format. As SADACS is a high-speed computer simulation, many analyses and subsequent animations can be done in a short period of time.

Structural Characteristics

For multi-flexible body structures where the bodies may move through large angles with respect to each other, the structural characteristics can change dramatically. Analysis of a four-body spacecraft shows significant change in the structural modes of concern when the solar array is rotated through 90 degrees. Open-loop transfer functions in figure 5 show this result.

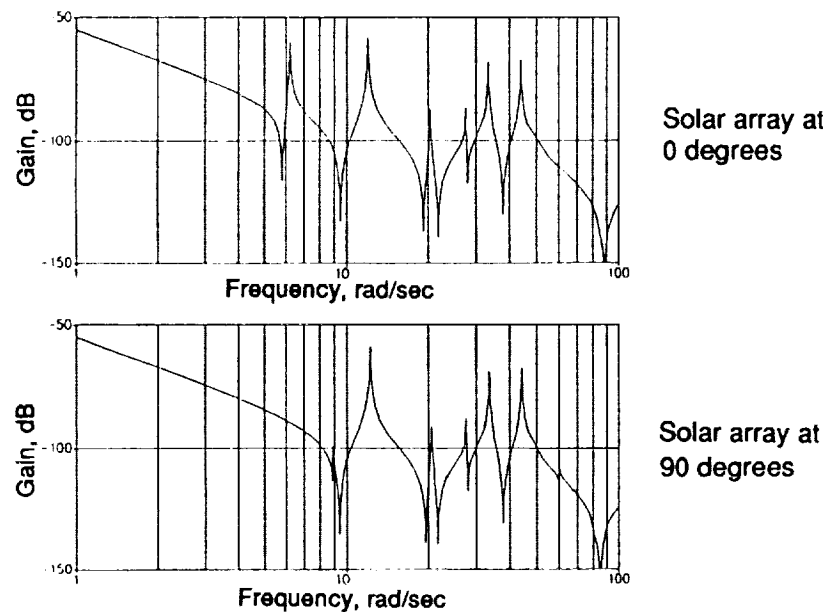


Figure 5. Open-Loop Transfer Functions for a Spacecraft with Different Solar Array Positions

The system mode frequencies and damping have been seen to change up to 50 percent. In addition to the frequency change, modes may swap with other modes or move to a different degree of freedom. The location of deformations in the structure may also change depending on the orientation of the bodies. To further complicate the picture, control forces and torques also affect how the structure behaves. The control system designer relies on knowledge of how the structural modes change in order to "tune" the control system for maximum performance. Animation of the multi-flexible body system is an extremely useful tool for visualizing all of these effects.

Animation and Control System Design

The FORMS animation software allows the control system analyst to employ visualization during the design process. Control system parameters can be changed and then animated to show whether flexibilities are suppressed or excited. This increased knowledge of the structural characteristics gives the designer more accuracy in determining, for example, optimum size and placement of sensors and actuators. Valuable insights are being realized now from current usage of animation software for control system design. Control system design is only one of many applications for multi-flexible body animation software.

-
- 1 Copyright 1991, Center for Simulation and Design Optimization of Mechanical Systems, University of Iowa.
 - 2 Copyright 1989, Center for Simulation and Design Optimization, University of Iowa.
 - 3 Smith, B., Rinaudot, G., Reed, K., Wright, T, Initial Graphics Exchange Specification (IGES) Version 4.0, U.S. Department of Commerce, June 1988.
 - 4 Copyright 1988, Brigham Young University.
 - 5 Copyright 1991, Computer Aided Design Software, Incorporated.
 - 6 Feng, H. and Pavlidis, T., "Decomposition of Polygons Into Simpler Components: Feature Generation for Syntactic Pattern Recognition," IEEE Transactions on Computers, June 1975, pp. 636-650.
 - 7 Copyright 1990, Open Software Foundation.
 - 8 Copyright 1988, Massachusetts Institute of Technology.
 - 9 Copyright 1990, Silicon Graphics, Incorporated.
 - 10 Sing, R. and VanderVoort, R., "Dynamics of Flexible Bodies in Tree Topology - A Computer-Oriented Approach," J. Guidance, Vol. 8, No. 5, pp. 584-590, Sept-Oct 1983.
 - 11 Bodley, C., Devers, A, Park, A, and Frisch, H., "A Digital Computer Program for Dynamic Interaction and Simulation of Controls and Structures (DISCOS)," NASA Tech. Paper 1219, May 1978.
 - 12 Jacot, A. D., Jones, R. E., and Juengst, C., "High Speed Simulation of Flexible Multibody Dynamics," Proceedings NASA Workshop on Structural Dynamics and Critical Interactions of Flexible Structures, Huntsville, Alabama, April 1986.
 - 13 Jones, R. E, Morse, T. W., and Russell, W. C., "High Speed Simulation of Multi-Flexible-Body Systems with Large Rotations," Proceedings AIAA Dynamics Specialist Conference, Monterey, California, April 1987.
 - 14 SD/EXACT is a trademark of Symbolic Dynamics, Inc., 928 Wright Avenue, Suite 108, Mountain View, California 94043.

